MULTIFAULT ISOLATION IN LINEAR NETWORKS BY THE METHOD OF JOINT SIGNATURE

Gary Lamour Long

NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

Multifault Isolation in Linear Networks
by the Method of Joint Signature

by

Gary Lamour Long

Thesis Advisor: S. G. Chan

December 1971

Approved for public release; distribution unlimited.

Multifault Isolation in Linear Networks
by the Method of Joint Signature

by

Gary Lamour Long
Lieutenant, United States Navy
B.S., Naval Postgraduate School, 1970

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
December 1971

ABSTRACT

A study was made to isolate one and two component faults in simple linear networks by the method of joint signature of order two whereby two measurements of a failed network are coded to all possible combinations of two component values which will produce that fault with all other elements at nominal values. Two additional measurements were obtained, a new joint signature of order two generated, and the faulty component set was selected as the fault pair with the same values at both joint signatures. The method involves numerous solutions of sets of nonlinear equations. By assuming many values of a failed network function the equations were presolved on a one time only basis for a network. One and two component faults were simulated in two simple low pass filter networks. The highest success rate in isolating faults was approximately 80%.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

Foremost, I wish to thank Professor Shu-Gar Chan for his patient guidance throughout the period of this study and for his suggestion of the thesis topic. I also wish to thank Lieutenant Jim Charles for reading the first drafts and Associate Professor Ralph Bach for his assistance and suggestions concerning the final draft. The services of Mrs. Helen Sutherland in typing the paper are sincerely appreciated.

I. INTRODUCTION

A. HISTORICAL BACKGROUND

1. Single Fault Isolation

Although there has long been an interest in fault isolation procedures, efforts in devising fault detection techniques have generally lagged behind other aspects of network theory. With the increasing sophistication of modern electronic systems came the progressively more complex trouble shooting procedures which are embodied in the automatic and semi-automatic test equipments, basically using one electronic system to check the performance of another. Although effective, these methods have proved to be expensive both monetarily and in the consumption of engineering expertise since generally test equipment has been designed to check only a specific system or specific type of network. It is against this background that some recent developments in fault isolation practices have attracted new attention, not so much because of the encouraging results obtained from experimental application as from the promise of a general fault isolation procedure applicable to a large class of electrical networks. The techniques developed thus far are still restrictive, applying only to single faults in linear networks, but form the nucleus of an emerging field of fault isolation theory.

a. The Proposal of Fault Signatures

Seshu and Waxman [1] were the first to propose a system of fault isolation by matching each element in a linear network with the signature of faults that the particular element could produce. Then by taking several measurements of a failed network and determining the fault signature, one could presumably detect the faulty element by matching

fault signatures. Seshu and Waxman proposed using test frequencies below the lowest nonzero break frequency, at least one above the highest finite break frequency and several test frequencies near complex poles and zeros. Their fault signatures were based solely upon the magnitude of the network function. In an analysis of an eight element one transistor network, where the transistor was replaced by the hybrid parameter model, Seshu and Waxman found that 41 of 62 computed fault signatures identified the faulty component uniquely while 15 others isolated the fault to two components. No experimental results were reported.

b. Experimental Results Using Fault Signatures

Stahl, Maenpaa and Stehman [2] have reportedly been 78% to 98% suscessful in isolating single element faults to a single component in linear networks of 8 to 15 components. Utilizing a computer program for fault diagnosis as proposed by Seshu and Waxman, they generated signatures and test frequencies for single element variations. The number of test frequencies used was three plus one for each complex breakpoint frequency between extreme critical frequencies. The network model employed was a two-port and the input impedance and voltage transfer ratio were the measured network functions. It is perhaps of importance to note that the approach of the two mentioned groups of authors does not make direct use of the bilinear nature of a linear network function F(s) to variations in a circuit element's value as presented by DeClaris [3]. By utilizing the bilinear property, which of course requires phase measurements, one can ascertain approximate values of a faulty component which can be used to choose between two or more possible faulty components.

2. Multiple Faults

Phung and Chan [4] have proposed an extension to fault isolation theory to include multiple faults in linear networks. The concept of a joint signature of order K was introduced whereby those combinations of K single element faults, i.e., K faulty elements, causing a network function F(s) to lie within some small area of the F(s) plane were coded with a signature. Then, by taking measurements of F(s), one could search the sets of elements with joint signature corresponding to the measured value of F(s) to determine the faulty set.

B. GOAL OF FAULT ISOLATION TECHNIQUES

The ultimate goal of fault isolation theory is to produce procedures whereby any number of faults in any type of network can be isolated with a minimum number of measurements and requiring little knowledge of circuit theory on the part of the individual isolating the faults. This means that a technician, by taking a few simple measurements, should be able to pinpoint faulty network components in all types of networks. In addition, the procedure should be capable of being automated.

II. THE PROBLEM

A. APPROACH

In this paper the concept of joint signature of order K, presented by Phung and Chan, is extended, by heuristic mathematical concepts, to include approximations of faulty component values and to include a mathematical basis for determing the number of test measurements required. Two simple example problems were treated by generating fault signatures and corresponding network function values via digital computer applications.

B. MULTIFAULT ISOLATION BY JOINT SIGNATURE

A method will be described whereby one can determine the approximate values of faulty network parameters. The term parameters is used to include all variables, except frequency or time, which appear in the mathematical description of the network function to describe the network input response relationship. By comparing determined values to nominal falues, faults may be isolated to any number of parameters. Since approximate parameter values are to be determined only one network function need be measured. In concept the network may either be linear or nonlinear requiring only that the output response to a given input be describable in either the time or frequency domain by a set of simultaneous nonlinear equations in which the unknowns are a set of network parameters. However, the following discussion assumes a linear network and further assumes that the network parameters are frequency independent such as would be the case if the network function were expressed in terms of elements once the complex frequency s was treated as a known constant.

Consider a network function $T(s,\underline{x})$ which is a ratio of polynomials in s and the vector $\underline{x}$ of all network parameters. Assume that there are N such parameters. Then, knowing that the network is faulty, N measurements are required to obtain N equations to solve for the network parameters. There are N unknowns. However, by assuming that there are a maximum of K faulty elements, the number of unknowns may be reduced.

The joint signature of order K is now defined as any possible combination of K parameter values which, with the remaining N-K parameters at nominal values, will cause the network function to have K specific values. In regard to a linear network, network function is now defined as a real equation such as $|T(s,\underline{x})| = 1.0$, $\underline{/T}(s,\underline{x}) = 90°$ or Imaginary part of $\{T(s,\underline{x})\} = 0.6$. In general there are $\binom{N}{K} = \frac{N!}{K!\,(N-K)!}$ such sets of K element values hereafter referred to as possible fault sets. These $\binom{N}{K}$ possible fault sets are all combinations of K element values, N-K remaining at nominal, which satisfy a set of K nonlinear equations. Note that not all members of a possible fault set need have a value different from nominal. Maintaining the assumption of a maximum of K faults let one take K measurements, determine the joint signatures of order K corresponding to the measured values and obtain the possible fault sets. By eliminating one previous measurement (equation) and obtaining a new one, a new set of possible fault sets is obtained with new joint signatures of order K. Borrowing from the theory of linear algebra, if there are K unknowns or K faults, an excess equation exists and th overall set of K+1 equations will be consistent for only one possible fault set (assuming linearity), the actual set of K faulty elements. The actual fault set could be determined by comparing the possible fault sets from the first K equations with those from the second K equations and selecting the possible fault set

common to both sets of equations. Two theoretical problems exist. First, the equations are nonlinear so that there may be more than one common possible fault set with two distinct joint signatures of order K. To reduce resulting ambiguities additional excess equations may be obtained. In fact, the second set of K equations may be entirely different from the first. The second problem, the assumption of a maximum of K faults, is more important. If the assumption is not true the procedure will fail but its failure may be detected by the troubleshooting technician. If there is no common possible fault set corresponding to the joint signatures, one can see that there are actually more than K network faults. Since the network equations are nonlinear, a second excess equation is required to reduce ambiguities if more than K faults are to be detected. Thus the choice of K is critical, a compromise between the complexity of calculations and the depth of fault isolation desired. The procedure for multifault isolation is now summarized below.

Step 1. Choose the maximum number of faults K to be considered.

Step 2. Take K network measurements to obtain K equations.

Step 3. Solve the K equations for the $\binom{N}{K}$ possible fault sets associated with the joint signature of order K.

Step 4. Eliminate one or more equations and take new measurements to obtain a new set of K equations.

Step 5. Solve the new equations for $\binom{N}{K}$ possible fault sets.

Step 6. Choose the common possible fault set from steps 3 and 5 as the fault set.

Step 7. If one desires to detect the presence of more than K faults, repeat the procedure. Compare the two fault sets from Step 6. If the two fault sets are different, there are more than K network faults.

C. PARAMETER PRESOLUTIONS

None of the proceding discussion is of importance to the network analyst diagnosing faults who does not have access to a digital computer unless, in some manner, the sets of simultaneous non-linear equations can be presolved. This can be accomplished by taking typical output values of a failed network and using these typical values as approximations to the actual measured values.

As an example consider a linear network function $T(s,\underline{x})$. Suppose $|T_o(s,\underline{x})|$ is the nominal output magnitude for a specified input and $\underline{/T_o(s,\underline{x})}$ is the nominal phase relationship. By taking areas of interest about the nominal values and dividing this region into cells, one can presolve the equations based on the values $|T(s,\underline{x})|$ and $\underline{/T(s,\underline{x})}$ at the centers of these cells. Then, when actual circuit measurements are taken and plotted into a particular set of cells, these cells will be coded to

a set of faulty parameters whose value has previously been determined by the procedures established in the preceding section. The complex calculations need be done only once for any network. Further, calculations may be performed on an impedance and frequency normalized model and the parameter values can merely be unscaled to apply to any original network.

III. EXPERIMENTAL PROCEDURE

A. PROBLEM ONE

1. Network

The network chosen for example one was a third order low pass Chebyshev filter with cut off frequency at 2500 radians per second and source resistance as shown in Figure 1. The network function used was the voltage transfer function with components as the parameters.

2. Number of Faults and Test Frequencies

Only one and two component faults were considered here. Test frequencies were selected at either side of the cut off frequency at 1500 and 3500 radians per second. The cut off frequency of 2500 radians per second was also used. The codes A, B, and C were used to identify the three frequencies 1500, 2500, and 3500, respectively. After it was determined that the use of a third test frequency did not appreciably affect results, the test frequency 2500 was discarded.

3. Problem Limitations

In order to limit the scope of the problem only variations in the real part of $T(s,\underline{x})$ of $\pm 22.5\%$ or less of the magnitude of the nominal $T(s,\underline{x})$ were considered at each test frequency. Simulated failed network function values which fell outside this region were discarded. The imaginary part was similarly restricted. The cell size chosen in the complex $T(s,\underline{x})$ plane was d units wide (real part) and d units high (imaginary part) where $d = 0.05 \cdot |T_o(s,\underline{x})|$ at the frequency of concern. The real and imaginary parts of $T(s,\underline{x})$ were considered, rather than magnitude and phase, to ease bookkeeping problems. The value of d was

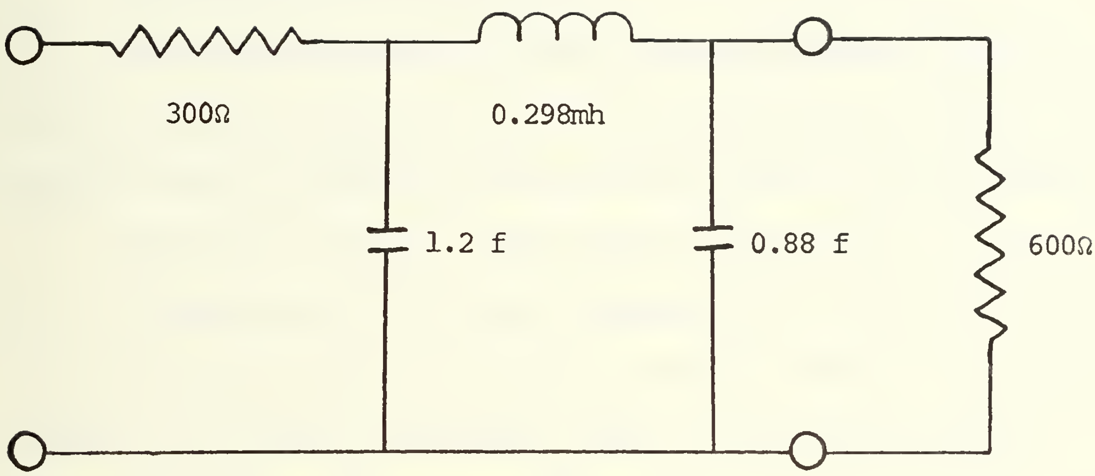

FIGURE ONE

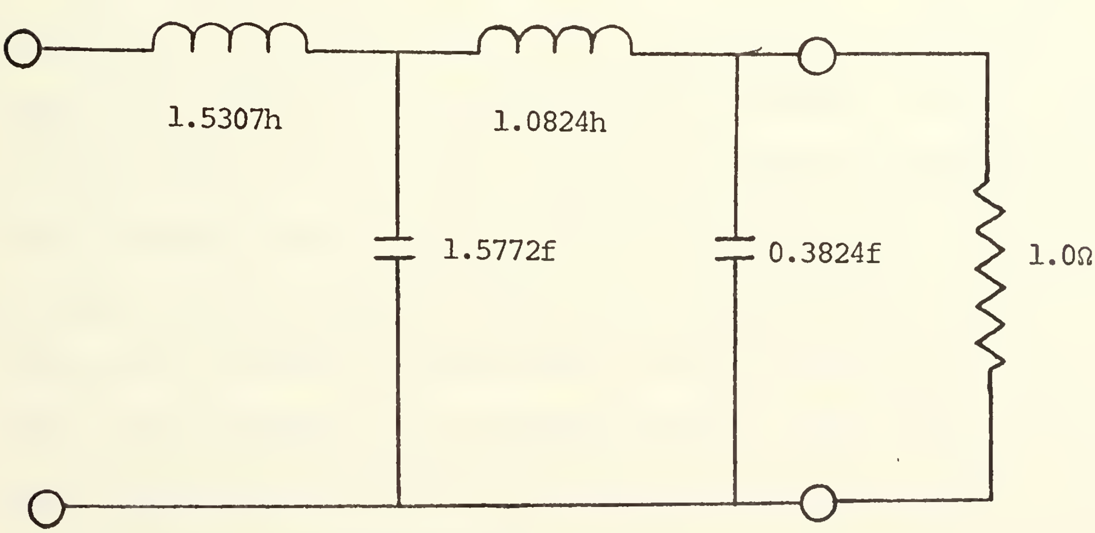

FIGURE TWO

selected as a reasonably small cell size compatible with approximation and yet within the accuracy of standard measuring equipment. The cells were then numbered as in Figure 3 where A is coded to $s = j1500$.

4. Generating Joint Signatures

The joint signatures of order two were generated by solving the two nonlinear equations below for all ten possible combinations of the five network elements and for each cell center at each frequency.

Real part of $T(s,\underline{x})$ = Real part of center of cell ij.

Imaginary part of $T(s,\underline{x})$ = Imaginary part of center of cell ij.

The equations were solved utilizing an earlier version of the computer program at the end of this thesis with a 73% success rate when an accuracy of 0.1% was required on $|T(s,\underline{x})|$.

5. Fault Definition and Fault Set Selection

Numerous one and two element variations were simulated and values of $T(s,\underline{x})$ computed. These values of $T(s,\underline{x})$ were then plotted at both test frequencies. If at either test frequency the value of $T(s,\underline{x})$ plotted fell outside the cell whose center was the nominal value of $T(s,\underline{x})$ then a fault was considered to exist. For example, a value of $T(s,\underline{x})$ at $s = j1500$ might lie in cell 37A while at $s = j3500$ $T(s,\underline{x})$ might lie in cell 64C for the same element variations. Then all ten possible fault sets with joint signature of order two corresponding to cell 37A were compared to the possible fault sets with joint signature of order two corresponding to cell 64C. The actual fault set was selected by choosing the element pair with the smallest percentage variation (from frequency A to C) from the nominal value, based on the element from each pair which had the largest percentage variation. Once the fault pair was determined the fault set was obtained by averaging

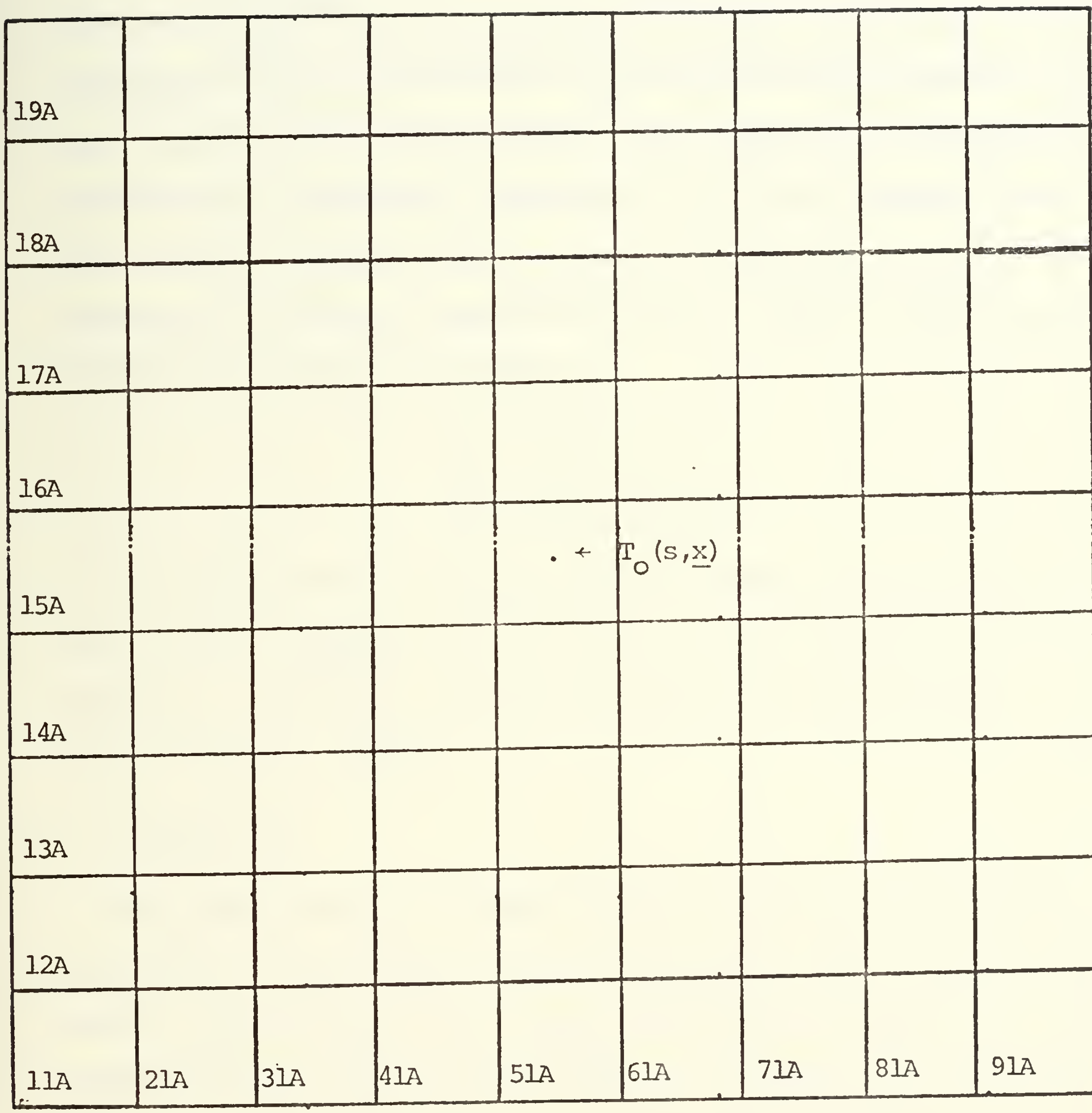

FIGURE THREE

the values of the fault pair at the two frequencies. Consider the element pair x_i and x_j. Let x_{iA}, x_{jA} be the possible fault set of these two elements at frequency A, etc. Then for each of the ten element pairs the percentage variation from the nominal value for each element was computed as $\left|\frac{x_{iA} - x_{iC}}{x_{io}}\right| \times 100$ where x_{io} is the nominal value of x_i. Of each element pair only the element with the largest percentage variation (the most deviant element) was considered. Then among the ten element pairs the 10 values of percentage variation for the 10 most deviant elements were compared. The fault pair was selected as the pair whose most deviant element had the smallest percentage variation. After the fault pair was determined the fault set was taken as:

$$x_{iF} = (x_{iA} + x_{iC})/2$$

$$x_{jF} = (x_{jA} + x_{jC})/2.$$

It can be seen that this decision criteria depends on the logarithmic sensitivities, $\frac{\partial T(s,\underline{x})}{\partial \ln x_i}$, being of the same order of magnitude for all elements i. Otherwise, fault selection favors the most sensitive (logarithmically) elements.

B. PROBLEM TWO

The network chosen for problem two was a normalized fourth order butterworth low pass filter as shown in Figure 2. Test frequencies used were $s_A = j0.5$ and $s_B = j1.2$. Again only one and two element faults were considered and a 5% cell size was used. The limitation of ±22.5% variation in $T(s,\underline{x})$ was retained. The program used to generate cells and joint signatures was that at the end of this thesis. The programs were run on the Naval Postgraduate School's IBM 360/67. The execution times were approximately 13 minutes for each test frequency. Since there were

10 element pairs and 81 cells at each frequency, this is about one second per solution of a set of equations. In the execution phase the program requires 66,000 bits of storage. The program is not presented as an example of efficient programming nor are any claims made as to the class of problems it will handle. However, in problem two the program was 94% successful in obtaining a solution to the sets of equations. Again required accuracy was 0.1% of $|T(s,\underline{x})|$. The fault set selection criteria were as in problem one. Attempts were made to normalize x $\Delta x_i = x_{iA} - x_{iB}$ by multiplying by the absolute value of the nonlogarithmic sensitivity, $\left|\frac{\partial T(s,\underline{x})}{\partial x_i}\right|$. If one lets $dT(s,\underline{x}) = \frac{\partial T(s,\underline{x})}{\partial x_i} \cdot \Delta x_i$ and selects the fault pair on the vasis of the smallest value $|dT(s,\underline{x})|$, this is seen to be a first order correction. The procedure heavily favored the least sensitive elements and was abandoned in favor of the previous selection criteria.

IV. DATA

A. PROBLEM ONE

1. Summary of Data for One and Two Component Faults

In order to minimize misrepresentation the data is presented in tabular form. Only element values other than those nominal ones have been listed and so care must be exercised in interpretation since the method of joint signature of order K always produces K "faulty" parameters even though some parameters may not be faulty at all. For instance, test No. 110 lists $x_2 = 1.59$, $x_5 = 591$ as a fault set while only x_2 is an actual faulty element. Inspection reveals that the nominal value of $x_5 = 600$ ohms and so the fault value is less than 1% from nominal or no fault at all. In an effort to simplify interpretation, results have been labeled as successful, partially successful and failure. The only true measure of success must be the return of the network to a non-faulty condition when the indicated faulty elements are replaced by components at nominal value. Accordingly, success has been defined as the case where the indicated fault pair agreed with the actual fault pair regardless of values except for very small differences previously discussed. Partial success means that a determined fault set agreed with the actual fault set as far as the element with largest percent variation was concerned. Program failure indicates that the computer program did not solve for the joint signature of order K of the actual fault set at one or both test frequencies. In these cases the selection of a fault set could not possibly have been correct since needed information was missing.

A summary of the results for problem one are listed below.

Number of Tests	Successful	Partially Successful	Program Failures
44	22	7	6

Number of Failures
15

2. Table of Data for One and Two Component Faults

TEST #	CELLS	DETERMINED FAULT SET*	ACTUAL FAULT SET*	SUCCESS	PROGRAM FAILURE
2	73A/38C	X3 = 293 X5 = 795	X5 = 800	YES	NO
8	63A/48C	X2 = 1.26 X5 = 763	X4 = 0.90 X5 = 750	YES	NO
9	45A/76C	X4 = 0.97 X5 = 574	X4 = 0.98	YES	NO
10	54A/77C	X4 = 0.96 X5 = 642	X4 = 0.98 X5 = 650	YES	NO
23	45A/66C	X1 = 309 X4 = .95	X4 = 0.95	YES	NO
45	65A/34C	X4 = 0.80 X5 = 633	X3 = 280	NO	NO
46	76A/13C	X3 = 257 X5 = 594	X3 = 260	YES	NO
50	45A/76C	X4 = 0.98 X5 = 574	X3 = 325	NO	NO
58	25A/96C	X3 = 352 X5 = 588	X3 = 352	YES	NO
59	46A/64C	X3 = 302 X5 = 532	X5 = 540	YES	NO
70	35A/85C	X3 = 358 X4 = 0.78	X3 = 325 X5 = 575	PARTIAL	NO
75	31A/98C	X3 = 334 X5 = 708	X3 = 360 X5 = 800	YES	NO
76	83A/17C	X4 = 0.79 X5 = 812	X3 = 280 X5 = 800	PARTIAL	NO
77	74A/26C	X3 = 279 X5 = 710	X3 = 285 X5 = 700	YES	NO
78	33A/97C	X1 = 269 X4 = 1.15	X3 = 352 X5 = 700	NO	NO
79	45A/74C	X3 = 344 X4 = 0.77	X3 = 352 X4 = 0.75	YES	NO

*X1 and X5 are in ohms, X3 in millihenries, X2 and X4 in microfarads.

TEST #	CELLS	DETERMINED FAULT SET	ACTUAL FAULT SET	SUCCESS	PROGRAM FAILURE
81	55A/56C	X2 = 1.22 X5 = 618	X3 = 280 X4 = 0.95	NO	NO
90	44A/78C	X2 = 1.47 X5 = 660	X2 = 240 X4 = 1.35	NO	YES
92	55A/77C	X1 = 316 X4 = 0.91	X3 = 280 X4 = 1.05	NO	NO
97	55A/67C	X4 = 0.92 X5 = 630	X3 = 250 X4 = 1.20	NO	NO
102	65A/31C	X1 = 290 X3 = 301	X2 = 0.95	NO	NO
110	35A/98C	X2 = 1.59 X5 = 591	X2 = 1.60	YES	NO
118	35A/88C	X2 = 1.54 X5 = 606	X2 = 1.54 X5 = 615	YES	NO
119	56A/64C	X3 = 292 X5 = 539	X5 = 560	YES	NO
121	74A/13C	X2 = 0.96 X5 = 707	X2 = 0.95 X5 = 680	YES	NO
125	45A/67C	X2 = 1.36 X5 = 624	X2 = 1.35 X5 = 625	YES	NO
127	56A/63C	X2 = 1.12 X5 = 528	X2 = 1.15 X5 = 525	YES	NO
131	37A/94C	X3 = 320 X5 = 477	X2 = 1.33 X5 = 450	PARTIAL	NO
132	36A/87C	X1 = 361 X3 = 318	X2 = 1.15 X5 = 525	NO	NO
137	47A/51C	X3 = 275 X5 = 465	X2 = 1.90 X4 = 0.50	NO	YES
139	45A/66C	X3 = 314 X5 = 595	X2 = 1.50 X4 = 0.78	NO	NO
141	34A/98C	X2 = 1.64 X5 = 631	X2 = 1.48 X4 = 0.95	PARTIAL	NO
142	44A/87C	X4 = 1.03 X5 = 611	X2 = 1.33 X4 = 0.98	NO	YES

TEST #	CELLS	DETERMINED FAULT SET	ACTUAL FAULT SET	SUCCESS	PROGRAM FAILURE
144	45A/76C	X4 = 0.97 X5 = 574	X2 = 1.60 X4 = 0.76	NO	NO
146	44A/87C	X4 = 1.02 X5 = 611	X2 = 1.15 X4 = 1.05	NO	YES
156	46A/69C	X2 = 1.33 X5 = 637	X2 = 1.90 X3 = 240	NO	YES
166	45A/65C	X3 = 311 X5 = 584	X2 = 1.15 X3 = 316	YES	NO
176	56A/66C	X1 = 335 X3 = 334	X1 = 344	YES	NO
180	64A/44C	X1 = 266 X3 = 294	X1 = 265	YES	NO
196	81A/27C	X2 = 1.16 X5 = 900	X1 = 265 X5 = 816	PARTIAL	NO
198	72A/13C	X2 = 1.08 X5 = 789	X1 = 220 X5 = 680	NO	YES
199	55A/56C	X1 = 307 X5 = 619	X1 = 318 X5 = 625	YES	NO
200	64A/32C	X1 = 257 X2 = 1.11	X1 = 235 X5 = 575	PARTIAL	YES
205	55A/69C	X2 = 1.30 X5 = 700	X1 = 360 X5 = 750	PARTIAL	NO

B. PROBLEM TWO

1. Summary of Data

The conditions for success and failure were as in problem one. Since the program successfully solved for 94% of the joint signatures, no program failures have been indicated. Instead, those few tests which involved program failures were completely ignored. A summary of the results of problem two are given below.

Number of Tests	Successful	Partially Successful	Failures
74	55	9	10

2. Table of Data

TEST#	CELLS	DETERMINED FAULT SET*	ACTUAL FAULT SET*	SUCCESS
3	54A/45B	X1 = 1.54 X5 = 1.06	X5 = 1.05	YES
5	56A/65B	X1 = 1.52 X5 = 0.94	X5 = 0.95	YES
6	67A/37B	X1 = 1.34 X5 = 0.87	X1 = 1.33 X5 = 0.85	YES
7	48A/75B	X4 = 0.35 X5 = 0.84	X5 = 0.85	PARTIAL
8	38A/95B	X2 = 1.68 X5 = 0.84	X2 = 1.63 X5 = 0.80	YES
9	49A/85B	X1 = 1.50 X5 = 0.80	X5 = 0.80	YES
12	62A/34B	X1 = 1.56 X5 = 1.20	X5 = 1.15	YES
13	52A/54B	X1 = 1.62 X5 = 1.16	X1 = 1.63 X5 = 1.15	YES
14	71A/34B	X1 = 1.54 X5 = 1.24	X5 = 1.25	YES
15	51A/53B	X1 = 1.68 X5 = 1.26	X1 = 1.68 X5 = 1.25	YES
20	53A/45B	X1 = 1.56 X5 = 1.08	X5 = 1.10	YES
21	34A/83B	X1 = 1.46 X5 = 1.18	X1 = 1.78 X5 = 1.10	YES
24	72A/34B	X4 = 0.36 X5 = 1.20	X5 = 1.20	PARTIAL
25	82A/25B	X1 = 1.88 X5 = 1.14	X1 = 1.47 X5 = 1.20	YES
35	49A/85B	X1 = 1.50 X5 = 0.80	X4 = 0.35 X5 = 0.80	PARTIAL

*X1 and X3 are in henries, X2 and X4 in farads, X5 in ohms.

TEST #	CELLS	DETERMINED FAULT SET	ACTUAL FAULT SET	SUCCESS
38	54A/46B	X1 = 1.43 X2 = 1.67	X4 = 0.46 X5 = 1.05	NO
39	52A/37B	X4 = 0.51 X5 = 1.12	X4 = 0.51 X5 = 1.15	YES
36	67A/63B	X4 = 0.20 X5 = 0.95	X4 = 0.19 X5 = 0.95	YES
67	47A/95B	X2 = 1.64 X5 = 0.88	X2 = 1.68 X5 = 0.85	YES
70	55A/56B	X1 = 1.47 X2 = 1.62	X4 = 0.43	NO
71	45A/65B	X1 = 1.60 X4 = 0.42	X1 = 1.59 X4 = 0.43	YES
73	56A/47B	X2 = 1.48 X3 = 1.20	X2 = 1.48 X3 = 1.18	YES
74	57A/17B	X1 = 1.38 X5 = 0.91	X2 = 1.38 X3 = 1.19	NO
78	56A/48B	X2 = 1.46 X3 = 1.24	X2 = 1.44 X3 = 1.23	YES
79	54A/54B	X1 = 1.59 X5 = 1.10	X2 = 1.63 X3 = 1.03	NO
82	35A/74B	X1 = 1.68 X2 = 1.60	X1 = 1.69	YES
85	55A/45B	X1 = 1.50 X3 = 1.07	X1 = 1.50	YES
87	45A/74B	X1 = 1.65 X5 = 1.01	X1 = 1.67	YES
88	35A/94B	X1 = 1.66 X2 = 1.67	X1 = 1.67 X2 = 1.68	YES
89	55A/65B	X2 = 1.60 X3 = 1.08	X1 = 1.56	NO
91	65A/45B	X1 = 1.46 X4 = 0.35	X1 = 1.48	PARTIAL

TEST #	CELLS	DETERMINED FAULT SET	ACTUAL FAULT SET	SUCCESS
94	55A/77B	X3 = 1.16 X5 = 1.05	X3 = 1.18	YES
95	65A/36B	X1 = 1.42 X5 = 0.99	X1 = 1.42	YES
96	66A/89B	X1 = 1.38 X3 = 1.35	X1 = 1.42 X3 = 1.28	YES
97	75A/16B	X1 = 1.38 X2 = 1.55	X1 = 1.36	YES
103	26A/93B	X1 = 1.83 X5 = 1.00	X1 = 1.85	YES
105	45A/64B	X1 = 1.63 X2 = 1.57	X1 = 1.60	YES
107	45A/74B	X1 = 1.65 X5 = 1.01	X1 = 1.63	YES
109	66A/63B	X4 = 0.23 X5 = 0.98	X4 = 0.26	YES
111	65A/31B	X3 = 0.95 X4 = 0.30	X3 = 0.94 X4 = 0.34	YES
114	54A/23B	X2 = 1.58 X3 = 0.96	X3 = 0.96 X4 = 0.46	PARTIAL
117	34A/78B	X2 = 1.66 X4 = 0.56	X2 = 1.68 X4 = 0.57	YES
128	56A/96B	X3 = 1.29 X4 = 0.32	X3 = 1.24 X4 = 0.30	YES
129	56A/76B	X3 = 1.29 X4 = 0.32	X3 = 1.18 X4 = 0.33	YES
130	55A/68B	X1 = 1.48 X3 = 1.14	X3 = 1.13	YES
132*	66A/63B	X4 = 0.23 X5 = 0.98	X4 = 0.25	YES

*This fault is almost identical to that of Test #109.

TEST #	CELLS	DETERMINED FAULT SET	ACTUAL FAULT SET	SUCCESS
134	76A/63B	X3 = 1.06 X4 = 0.18	X4 = 0.20	YES
136	76A/62B	X2 = 1.59 X4 = 0.12	X4 = 0.15	YES
137	76A/53B	X1 = 1.49 X4 = 0.18	X2 = 1.56 X4 = 0.18	YES
138**	76A/63B	X3 = 1.06 X4 = 0.18	X4 = 0.18	YES
140	65A/54B	X4 = 0.30 X5 = 1.02	X4 = 0.30	YES
142	55A/54B	X1 = 1.55 X3 = 1.06	X4 = 0.33	NO
156	44A/38B	X1 = 1.45 X4 = 0.57	X1 = 1.48 X4 = 0.57	YES
153	34A/49B	X4 = 0.61 X5 = 1.02	X4 = 0.63	YES
155	34A/48B	X4 = 0.59 X5 = 1.02	X4 = 0.57	YES
157	45A/57B	X2 = 1.54 X4 = 0.53	X4 = 0.52	YES
158	54A/18B	X1 - 1.39 X2 = 0.52	X1 - 1.39	NO
159	45A/56B	X4 = 0.46 X5 = 0.98	X4 = 0.48	YES
164	55A/56B	X1 = 1.51 X3 = 1.10	X4 = 0.43	NO
165	25A/84B	X1 = 1.70 X2 = 1.66	X1 = 1.75 X4 = 0.43	PARTIAL
167***	35A/74B	X1 = 1.68 X2 = 1.60	X1 = 1.68 X4 = 0.40	YES

**See Test #134

*** See Test #82

TEST #	CELLS	DETERMINED FAULT SET	ACTUAL FAULT SET	SUCCESS
168****	55A/54B	X1 = 1.55 X3 = 1.06	X4 = 0.34	NO
169*****	45A/74B	X1 = 1.65 X5 = 1.01	X1 = 1.62 X4 = 0.34	PARTIAL
171	75A/16B	X1 = 1.38 X2 = 1.55	X1 = 1.34 X4 = 0.34	PARTIAL
203	57A/49B	X2 = 1.40 X3 = 1.32	X2 = 1.40 X3 = 1.28	YES
205	44A/95B	X2 = 1.74 X3 = 1.13	X2 = 1.68 X3 = 1.14	YES
206	54A/63B	X2 = 1.68 X3 = 0.95	X2 = 1.70 X3 = 0.99	YES
208	56A/87B	X2 = 1.54 X3 = 1.22	X2 = 1.56 X3 = 1.24	YES
209	57A/48B	X2 = 1.41 X3 = 1.30	X2 = 1.42 X3 = 1.24	YES
210	57A/18B	X1 = 1.35 X5 = 0.86	X2 = 1.35 X3 = 1.24	NO
212	54A/11B	X2 = 1.58 X3 = 0.93	X3 = 0.89 X4 = 0.45	PARTIAL
216	66A/64B	X2 = 1.54 X4 = 0.31	X3 = 1.11 X4 = 0.27	YES
217	35A/79B	X3 = 1.21 X4 = 0.56	X3 = 1.15 X4 = 0.55	YES
221	62A/88B	X3 = 1.40 X5 = 1.22	X3 = 1.32 X5 = 1.20	YES

****See Test #142

*****See Test #107

V. CONCLUSIONS AND DISCUSSION OF RESULTS

A. CONCLUSIONS

The method of multifault isolation presented in this paper is a workable procedure for isolating faults in linear networks. The degree of success attained in problem two represents a realistic starting point for achieving high success rates in relatively complex networks. Inspection of the data for problem two revealed that of the 10 failures, 5 were associated with tests in which the network had failed at only one frequency. In problem one, 4 of the 16 failures occurred when the network had failed at only one frequency. This indicates that if one is to detect very small drift failures, then the cells may have to be very small near the nominal values.

The success rates mentioned for joint signature were for a single solution to a set of equations. No multiple solutions were obtained but the lack of multiple solutions was not considered an important deficiency. No claim is made as to the distribution of the simulated network faults.

B. ADVANTAGES OF THE METHOD OF JOINT SIGNATURE

The foremost advantage of the isolation procedure by joint signature is its simplicity of application. Although it may take considerable effort to establish the joint signatures and to code particular cells to a fault set, this is required only once for any network. The fault diagnoser need only take and plot simple measurements. The network trouble-shooter need have little knowledge of circuit theory or even of the particular circuit being tested. When equations of magnitude have been

utilized the equipment required may only be a signal generator and a volt meter. The procedure is general and may be applied to any linear network with possible applications to nonlinear systems.

C. A QUESTION OF FEASIBILITY

The problems encountered in solving simultaneous nonlinear equations and the large execution times required for complex networks may not allow the approach to be feasible. Certainly the procedure, because of the effort required to generate joint signatures, is limited to circuits which have wide use. One method to reduce problem complexity is to express the network function in terms of some comprehensive parameters rather than elements. The best isolation that could be obtained would then be to some group of elements or subnetwork. Utilizing the conventional two-port parameters would require a prohibitive number of variables (8 real) to describe a subnetwork assuming cascade two-port connections. Furthermore, if the parameters are not frequency independent then one is required to somehow obtain several network equations at one frequency.

The simple networks studied each had only one break frequency and so the choice of test frequencies was obvious. In complicated networks one will be required to take at least one measurement near each break frequency. This may also serve to determine the number of faults which will be considered.

If the network output is either zero or infinite one cannot apply the method of joint signature without utilizing interior test points. Otherwise, the magnitude of faults which can be considered is unlimited. It is expected that catastrophic failures are more easily isolated and thus the cell size might be increased farther away from nominal value. Further improvement in results could perhaps be obtained from using an improved procedure for selecting the fault pair. As mentioned, the method used depended on the logarithmic sensitivities being of the same order of magnitude. However, the sensitivities of elements one, two and three in problem four were 10^4 times those of elements four and five revealing why element four faults were difficult to isolate.

VI. SUGGESTIONS

A. EXTENSION TO COMPLEX LINEAR NETWORKS

Several questions concerning the feasibility of fault isolation by joint signature have already been posed. In order to arrive at some answers it is suggested that the joint signature concept be studied for three and four element faults in very simple networks. Since the number of possible cell combinations is $M(M-1)(M-2)\ \ldots\ (M-Q)$, where M is the number of cells at any frequency and Q is the number of measurements, large cells will be required with possibly smaller cells toward nominal values. However, as one increases the number of faults considered, the number of possible fault sets decreases. If one could consider N faults in a network of N elements, there would be only one fault set. Methods of expressing a network function in terms of some few parameters and of obtaining network measurements to determine faulty parameters are needed. Two possible approaches are to load the network with a variable known load or to utilize interior test points to measure intermediate network functions in the same parameters. The study should also allow for the detection of more than K faults.

B. APPLICATION TO NONLINEAR NETWORKS

Simple nonlinear networks may be studied to determine fault set selection techniques and applicability.

```
C     THIS IS A FORTRAN IV G PROGRAM TO SOLVE UP TO THREE SIMULTANEOUS
C     COMPLEX EQUATIONS FOR UP TO SIX REAL VARIABLES.
C
C
C     THE PROGRAM LAYOUT IS DESIGNED FOR CIRCUIT ANALYSIS. THE
C     EQUATIONS ARE THE TRANSFER FUNCTIONS OF A LINEAR CIRCUIT AT
C     DIFFERENT FREQUENCIES.
C     THE EQUATIONS (REAL) ARE NON-LINEAR AND IN THE SIMPLEST CASE
C     ARE BIQUADRATIC. THIS PROGRAM ALLOWS THE DETERMINATION OF M
C     ELEMENT VALUES (REAL UNKNONNS) GIVEN M/2 COMPLEX EQUATIONS.
C
C     THE USER SUPPLIES ONE SUBROUTINE, SUBROUTINE CALCT, AND IN
C     ADDITION FURNISHES THE DATA DECK.
C
C
C
C     DATA FORMAT IS DESCRIBED SEPARATELY
      INTEGER P,PP,PA,PN
      COMPLEX S,T,CENT,Z,TNOM,SENX,SAVT
      DIMENSION ERRMUG(200),DELV(50),DXNORM(50),DELXN(6)
      DIMENSION XVAR(6),EPO(20)
      COMMON/ARRAY1/S(3),T(3),X(50)
      COMMON/ARRAY2/CENT(15,15,3),TNOM(3)
      COMMON/ARRAY3/PSAV(50)
      COMMON/ARRAY4/SENX(50)
      COMMON/ARRAY5/NVAR(6)
      COMMON/ARRAY6/Z(3)
      COMMON/ARRAY7/DELX(50)
      COMMON/ARRAY8/XSAV(50),SAVT(3)
C
C     SUBROUTINE READ READS IN ALL DATA EXCEPT PERMUTATIONS
C
      CALL READ(ICELL,IXLIM,IYLIM,NEQ,IMENT,NPERM,IVX,FSENS,ACC)
C     TMUG IS THE SUM OF THE MAGNITUDES OF THE NOMINAL TRANSFER FUNCTIONS
      TMUG=0.0
C     IPO IS AN INDICATOR TO SUBROUTINE PERM THAT PERMUTATION DATA
C     IS TO BE READ.
      IPO=0
C     CALCULATE TMUG
C     NEQ, INPUT DATA, IS THE # OF EQUATIONS.
      DO 17 IQ=1,NEQ
C     SUBROUTINE CALCT CALCULATES A VALUE OF THE TRANSFER FUNCTION
C     USING CURRENT ELEMENT VALUES AND THE FREQUENCY IN THE S ARRAY
C     DENOTED BY THE CALLING ARGUEMENT.
      CALL CALCT(IQ)
C     THE TNOM ARRAY WILL RETAIN THE NOMINAL TRANSFER FUNCTIONS
C     AT THE DIFFERENT FREQUENCIES.
      TNOM(IQ)=T(IQ)
```

```
      TMUG=TMUG+CABS(TNOM(IO))
   17 CONTINUE
C     SUBROUTINE SENSIT CALCULATES THE SENSITIVITY FUNCTIONS OF ALL THE
C     ELEMENTS OF THE TRANSFER FUNCTION.
C     IMENT, INPUT DATA, IS THE # OF ELEMENTS IN T(S).
C     FSENS IS THE FREQ AT WHICH SENSITIVITIES ARE CALCULATED AND
C     IS PART OF THE DATA INPUT.
      CALL SENSIT(FSENS,IMENT)
      WRITE(6,111)
  111 FORMAT('1',20X,'SENSTIVITY FUNCTIONS CALCULATED AT S=J')
      WRITE(6,112)FSENS
  112 FORMAT('+',59X,1F6.0,////)
      DO 113 LU=1,IMENT
      WRITE(6,114)
  114 FORMAT('0',20X,'SENSITIVITY OF X(   )=')
      WRITE(6,115)LU,SENX(LU)
  115 FORMAT('+',37X,1I2,2X,2E11.3)
  113 CONTINUE
C     ACC IS THE REQUIRED ACCURACY ON SOLUTIONS AND IS FURNISHED DATA.
C     ACC,AS USER SUPPLIED, IS A FRACTION OF TMUG. IN THE PROGRAM
C     IT IS CONVERTED TO A REAL NUMBER.
      WRITE(6,219)ACC
  219 FORMAT('0',20X,'ACCURACY LIMIT ON SOLUTION IS ',1E14.6)
C     SUBROUTINE CENTER CALCULATES THE TRANSFER FUNCTION AT THE CENTER
C     OF THE CELLS OF INTEREST.
C     ICELL, INPUT DATA, IS THE CELL SIZE AS A PERCENTAGE OF TMUG.
C     IXLIM, INPUT DATA, IS THE LIMIT ON THE REAL PART OF T(S) TO BE
C     CONSIDERED. EXPRESSED AS A PERCENTAGE OF TMUG.
C     IYLIM IS A LIMIT ON THE IMAGINARY PART.
C     IF AND IG ARE DO LOOP LIMITS CALCULATED IN SUBROUTINE CENTER
C     BASED ON ICELL,IYLIM,IXLIM.
      CALL CENTER(ICELL,IXLIM,IYLIM,NEQ,IF,IG)
C     INITIALIZE ERRORS TO ZERO
      ERRM=0.0
      ERMM=0.0
C     CONVERT ACC TO PURE NUMBER
      ACC=(ACC*TMUG)
C     NOW BEGIN THE MINIZATION PROCESS, THE STARTING POINT DEPENDS ON THE
C     NUMBER OF EQUATIONS NEQ
      IF(NEQ.EQ.1)GO TO 71
      IF(NEQ.EQ.2)GO TO 72
C     THE DO LOOPS GOING TO 18 DETERMINE THE VALUES OF T(S), UP TO 3,
C     OF INTEREST WHICH ARE STORED IN THE CENT ARRAY.THE FIRST DIMENSION
C     OF CENT GIVES THE REAL CELL NUMBER, THE 2ND GIVES THE IMAGINARY,
C     THE 3 RD THE FREQUENCY TO BE USED.
      DO 21 IR=1,IF
      DO 21 IS=1,IG
C     Z(3) IS THE CURRENT VALUE OF T(S) AT THE 3RD FREQUENCY
```

```
      Z(3)=CENT(IR,IS,3)
C     S(3) IS THE 3RD FREQUENCY , IN COMPLEX FORM
      F3=AIMAG(S(3))
C     THE ERROR OR OBJECTIVE FUNCTION IS THE MAGNITUDE OF THE
C     DIFFERENCE OF THE REQUIRED VALUE OF T(S) AND THE CURRENT VALUE
C     OF T(S) CALCULATED FROM CURRENT ELEMENT VALUES.
C     THE REQUIRED VALUES OF T(S) ARE IN THE CENT ARRAY AND ARE
C     TRANSFERED TO THE Z ARRAY.
      ERMM=(CABS(Z(3)-TNOM(3)))
 72   DO 22 J5=1,IF
      DO 22 J6=1,IG
      Z(2)=CENT(J5,J6,2)
      ERRM=(CABS(Z(2)-TNOM(2)))
      F2=AIMAG(S(2))
 71   DO 18 J7=1,IF
      DO 18 J8=1,IG
      Z(1)=CENT(J7,J8,1)
C     THE ERRMUG ARRAY CONTAINS THE VALUES OF THE OBJECTIVE FUNCTIONS
C     ON SUCCESSIVE ITERATIONS. OBJ FUNCTION AND ERROR TERM WILL BE USED
C      INTERCHANGEABLY.
C     THE 1ST ERROR TERM IS THE SUM OF THE MAGNITUDES OF
C     THE DIFFERENCE FROM NOMINAL T(S)'S AND DESIRED VALUES OF T(S)
      ERRMUG(1)=ERRM+ERMM+(CABS(Z(1)-TNOM(1)))
      F1=AIMAG(S(1))
      WRITE(6,116)
 116  FORMAT('1',25X,'////////////////////////////////////////////////////
     C/////////////////////////////////////')
      WRITE(6,117)
      WRITE(6,117)
 117  FORMAT(' ',25X,'/
     C                                    /')
      IF(NEQ.EQ.3)WRITE(6,121)
 121  FORMAT(' ',25X,'/      CELL #',9X,'FREQUENCY=',10X,'RPS',7X,'CENTER
     C=',25X,'/')
      IF(NEQ.EQ.3)WRITE(6,122)IR,IS,F3,Z(3)
 122  FORMAT('+',37X,1I2,1X,1I2,14X,1F9.2,18X,1F7.4,3X,1F7.4)
      IF(NEQ.EQ.2)WRITE(6,121)
      IF(NEQ.GE.2)WRITE(6,122)J5,J6,F2,Z(2)
      WRITE(6,121)
      WRITE(6,122)J7,J8,F1,Z(1)
      WRITE(6,117)
      WRITE(6,117)
      WRITE(6,131)
 131  FORMAT(' ',25X,'////////////////////////////////////////////////////
     C/////////////////////////////////////')
      WRITE(6,301)
 301  FORMAT('0',/)
C     SUBROUTINE PERM SELECTS A WORKING SET OF VARIABLES, TWICE THE
```

```
C     NUMBER OF EQUATIONS, FROM THE POSSIBLE  VARIABLES, THESE
C     ARE FURNISHED DATA.
C     NPERM IS THE NUMBER OF PERMUTATIONS.
      DO 20 JB=1,NPERM
      CALL PERM(IVX,JB,NPERM,IPO)
C     MUM IS AN INDICATOR TO LIMIT THE NUMBER OF ITERATIONS
      MUM=1
      IFG=1
      ERRSAV=ERRMUG(MUM)
      IF((J8.NE.IG/2+1).OR.(J7.NE.IF/2+1)) GO TO 63
      IF(NEQ.EQ.1) GO TO 61
      IF((J6.NE.J8).OR.(J5.NE.J7)) GO TO 63
      IF(NEQ.EQ.2)GO TO 61
      IF((IS.NE.J8).OR.(IR.NE.J7)) GO TO 63
      GO TO 61
 63   NOM=1
      DX=0.05
C     DX DETERMINES THE STEP SIZE FROM LOCAL MINIMUM
C     SIGN DETERMINES THE DIRECTION OF STEP IN CONJUNCTION WITH PN
C     PP, AND PA.
      SIGN=-1.0
      PN=1
      PP=0
      PA=0
      C=1.0
      ISTEP=2
C     RSAV ARRAY SAVES NOMINAL VALUES
C     XSAV ARRAY SAVES BEST ESTIMATES THUS FAR
      DO 171 LAN=1,IVX
      NAN=NVAR(LAN)
 171  XSAV(LAN)=RSAV(LAN)
   33 CONTINUE
      MAM=MUM
      MUM=MUM+1
      ERRMUG(MUM)=0.0
C     ERRMAG IS AN INTERIM ERROR VALUE AT ONLY ONE FREQUENCY
C     IFIG IS A COUNTER FOR DE-INCREMENTING.
      IFIG=1
      IF(MUM.GE.200)GO TO 49
C     DSUM IS CURRENT VALUE OF OBJECTIVE FUNCTION
      DSUM=ERRMUG(MAM)
C     NEWRAP PERFORMS MODIFIED NEWTON-RAPHSON METOD ON VARIABLES
      CALL NEWRAP(NEQ,DSUM,IVX,ACC,C)
C     CALCULATE NEW ERROR TERM USING NEW ESTIMATES.
 211  DO 19 IT=1,NEQ
      CALL CALCT(IT)
      ERRMUG(MUM)=ERRMUG(MUM)+CABS(T(IT)-Z(IT))
 19   CONTINUE
```

```
C      IS ERROR LESS THAN REQUIRED ACCURACY ?
C      IF SO WRITE ANSWERS
 87    IF(ERRMUG(MUM).LE.ACC)GO TO 25
C      IBACK IS AN INDICATOR THAT DE-INCREMENTING IS REQUIRED.
       IBACK=2
       IF(MUM.GT.10)ICOUNT=MUM-10
C      NEXT CARD DETECTS A BLOW UP.
       IF((ERRMUG(MUM).GT.ERRMUG(1)).AND.(MUM.GT.10))GO TO 69
C      INCREMENT WAS TOO LARGE. DE-INCREMENT.
 504   IF(ERRMUG(MUM).GT.ERRMUG(MAM)) IBACK=1
 245   IF(ERRMUG(MUM).GT.ERRMUG(MAM)) GO TO 54
       IF(MUM.LE.10)GO TO 611
C      TEST RATE OF MINIZATION TO DETECT A LOCAL MINIMA.
       IF(((ERRMUG(ICOUNT)-ERRMUG(MUM)).LT.0.05*ERRMUG(ICOUNT)))GO TO 69
C      MINIZATION TAKING PLACE, SAVE GOOD ESTIMATES
 611   ERRSAV=ERRMUG(MUM)
       DO 141 JUN=1,IVX
       KUN=NVAR(JUN)
 141   XSAV(KUN)=X(KUN)
       DO 161 LUN=1,NEQ
 161   SAVT(LUN)=T(LUN)
       GO TO 33
 61    DO 62 MUNG=1,NEQ
 62    T(MUNG)=TNOM(MUNG)
C      STATEMENTS 25 TO 66 PRINT OUT ANSWERS OR BEST ESTIMATES
 25    DO 40 IJK=1,IVX
       ILK=NVAR(IJK)
       DELV(ILK)=X(ILK)-RSAV(ILK)
       DXNORM(ILK)=DELV(ILK)*CABS(SENX(ILK))
       DELXN(IJK)=DXNORM(ILK)
       XVAR(IJK)=X(ILK)
 40    CONTINUE
       SOLN=1.0
       DO 59 K2=1,NEQ
       WRITE(6,43)
 43    FORMAT('0 ','MUM=',9X ,'DELXN(   )=',16X,'X(   )=',18X,'T(   )=',23X,
      C'ERROR=')
       WRITE(6,42)MUM,NVAR(K2),DELXN(K2),NVAR(K2),XVAR(K2),K2,T(K2),ERRMU
      CG(MUM)
 42    FORMAT('+',5X,1I4,11X,1I2,2X,1E11.3,7X,1I2,2X,1E11.3,9X,1I2,2X,1F7
      C.3,3X,1F7.3,12X,1E11.3)
 59    CONTINUE
       NAQ=NEQ+1
       DO 66 K9=NAQ,IVX
       WRITE(6,67)
 67    FORMAT('0',        14X ,'DELXN(   )=',16X,'X(   )=')
       WRITE(6,68) NVAR(K9),DELXN(K9),NVAR(K9),XVAR(K9)
 68    FORMAT('+',20X,1I2,2X,1E11.3,7X,1I2,2X,1E11.3)
```

```
 66    CONTINUE
       WRITE(6,301)
       IF(ISTEP.EQ.1)GO TO 70
       GO TO 32
C      SUBROUTINE TWOVAR PERFORMS RELAXATION METHOD ON VARIABLES
C      WHEN NEWTON-RAPHSON METHOD FAILS.
 49    CALL TWOVAR(ACC,SOLN,IVX,NEQ,ERROR)
       ERRMUG(MUM)=ERROR
       IF(SOLN.EQ.2)GO TO 25
       IF(ERROR.LE.(10.0*ACC)) GO TO 39
       GO TO 32
C      LAST INCREMENT WAS TOO LARGE, SUBTRACT PART OFF. CONTINUE THIS
C      PROCESS UNTIL ERROR INCREASES.
 54    ERD(IFIG)=ERRMUG(MUM)
C      ERD ARRAY IS TEMP STORAGE FOR ERROR TERMS WHILE DE-INCREMENTING.
  798  IFM=IFIG
       IFIG=IFIG+1
C      IF ERROR HAS INCREASED, ADD BACK LAST INC AND EXIT.
C      LIMIT IS 20 DE-INCREMENTS.
 613   IF(IFIG.GE.20)GO TO 33
       DO 616 KUNE=1,IVX
       P=NVAR(KUNE)
 616   X(P)=X(P)-.40*DELX(P)
       IFIG=IFIG+1
C      AFTER DE-INCREMENTING GO TO 79 AND CALC NEW ERROR TERM.
   79  ERD(IFIG)=0.0
       DO 624 MAN=1,NEQ
       CALL CALCT(MAN)
 624   ERD(IFIG)=ERD(IFIG)+CABS(Z(MAN)-T(MAN))
       IF(ERD(IFIG).LT.ERD(IFM)) GO TO 798
 752   DO 751 LAG=1,IVX
       P=NVAR(LAG)
 751   X(P)=X(P)+.40*DELX(P)
       ERRMUG(MUM)=ERD(IFM)
       DO 614 LULU=1,NEQ
 614   CALL CALCT(LULU)
C      END OF DE-INCREMENTING ROUTINE.
       GO TO 33
 39    WRITE(6,118)
 118   FORMAT('0        NO SOLUTION GENERATED. THE FOLLOWING VALUES ARE
      CTHE BEST ESTIMATES')
C      ERRMUG(MUM)=ERRSAV
       DO 151 MUN=1,IVX
       NUN=NVAR(MUN)
  151  X(NUN)=XSAV(NUN)
       DO 800 JYZ=1,NEQ
 800   T(JYZ)=SAVT(JYZ)
       GO TO 25
```

```
C      STATEMENTS 69 TO 81 STEP THE FUNCTION OUT OF A LOCAL MINIMUM
C      IN VARIOUS DIRECTIONS, DEPENDING ON THE NUMBER OF THE TRY, AND
C      ALSO GIVE A NEW STARTING POINT TO A SEARCH THAT HAS BLOWN UP.
C      THE MAGNITUDE OF THE STEP IS INCREASED EACH TIME A NEW STEP
C      IS TAKEN IN A PREVIOUSLY TRIED DIRECTION.
 69    NOM=NOM+1
       SOLN=1.0
       IF((NOM.EQ.2).OR.(NOM.EQ.3).OR.(NOM.EQ.4)) GO TO 53
       DXMAG=DX*FLOAT(PN)
       PP=PP+1
       PA=PA+1
       IF(PP.NE.IVX+1)GO TO 70
       SIGN=SIGN*(-1.0)
       PP=1
       ISTEP=1
C      IF(ERRMUG(MUM).LT.(((FLOAT(ICELL))*TMUG)/200.0)) GO TO 39
 70    P=NVAR(PP)
       IFG=IFG+1
       ISTEP=2
       IF(IFG.GT.20) GO TO 49
       IF(PA.EQ.(2*IVX))PN=PN+1
       IF(PA.EQ.(2*IVX))PA=0
       X(P)=RSAV(P)*(1.0+DXMAG*SIGN)
       ERRMUG(MUM)=0.
       DO 81 KOW=1,NEQ
       CALL CALCT(KOW)
       ERRMUG(MUM)=ERRMUG(MUM)+(CABS(T(KOW)-Z(KOW)))
   81  CONTINUE
       IF(ERRMUG(MUM).GT.ERRMUG(1))GO TO 69
       IM=MUM-1
       DO 89 ISN=ICOUNT,IM
  89   ERRMUG(ISN)=ERRMUG(MUM)*1.05
       GO TO 33
 53    C=C/3.0
       DO 55 KIN=1,IVX
       LIN=NVAR(KIN)
 55    X(LIN)=RSAV(LIN)
CC     MAYBE NEED TO USE RSAV
       DO 56 MIN=1,NEQ
 56    T(MIN)=TNOM(MIN)
CC     OR TNOM(MIN)
       ERRMUG(MUM)=ERRSAV
       GO TO 33
C      AFTER EACH SEARCH RESTORE VARIABLES TO THEIR NOMINAL VALUES
 32    DO 31 MM=1,IMENT
       X(MM)=RSAV(MM)
 31    CONTINUE
 20    CONTINUE
```

```
   21 CONTINUE
      IF(NEQ.EQ.1)STOP
  22  CONTINUE
      IF(NEQ.EQ.2)STOP
  18  CONTINUE
      STOP
      END


      SUBROUTINE NEWRAP(NEQ,DSUM,IVX,ACC,C)
      INTEGER P
      COMPLEX S,T,TSAV,Z
      DIMENSION TSAV(3),DF(2),D2F(6),DX(6),D1F(6)
      COMMON/ARRAY1/S(3),T(3),X(50)
      COMMON/ARRAY3/RSAV(50)
      COMMON/ARRAY5/NVAR(6)
      COMMON/ARRAY6/Z(3)
      COMMON/ARRAY7/DELX(50)
C     NVAR ARRAY CONTAINS SUBSCRIPTS OF CURRENT VARIABLES
C     SELECT, SEQUENTIALLY, THE VARIABLE TO BE INCREMENTED,SUBSCRIPT P
C     SAVE CURRENT TRANSFER FUNCTIONS VALUES
      DO 7 IJ=1,NEQ
      TSAV(IJ)=T(IJ)
  7   CONTINUE
C     CALC 1ST AND 2ND DERIVATIVES OF X(P)
      DO 29 L=1,IVX
      P=NVAR(L)
      DX(L)=1.E-2*RSAV(P)
      IF(DSUM.LT.10.0*ACC)DX(L)=0.01*DX(L)
  8   DO 101 M2=1,2
      DELTAF=0.0
      X(P)=X(P)+DX(L)
      DO 99 M1=1,NEQ
      CALL CALCT(M1)
      DELTAF=DELTAF+(CABS(T(M1)-Z(M1)))
  99  CONTINUE
      IF(M2.EQ.1)SAVE=DELTAF
      IF(M2.EQ.1)DF(1)=DELTAF-DSUM
      IF(M2.EQ.2)DF(2)=DELTAF-SAVE
 101  CONTINUE
C     IF 2ND DER IS ZERO AVOID DIVIDE CHECK ERROR
      IF(ABS(DF(2)-DF(1)).LE.1.E-9)DX(L)=DX(L)*10.0
      IF(ABS(DF(2)-DF(1)).LE.1.E-9)GO TO 8
   79 D2F(L)=(DF(2)-DF(1))/DX(L)**2
```

```
 83   D1F(L)=(DF(1)+DF(2))/(2.0*DX(L))
      FACT=1./C
C     NEAR SOLUTION LIMIT INCREMENT STEPS.
      IF(DSUM.LE.10.0*ACC)FACT=0.5/C
C     CALCULATE INCREMENT BY NEWTON-RAPHSON
      DELX(P)=(-D1F(L)/D2F(L))*FACT
C     LIMIT DELTA X(P) TO 0.2*(MAG OF X(P))
      IF(ABS(DELX(P)).GT.(0.1*ABS(X(P))))DELX(P)=0.1*DELX(P)*ABS(X(P)/DE
     CLX(P))
      X(P)=X(P)-2.0*DX(L)
 29   CONTINUE
      DO 31 N=1,IVX
      P=NVAR(N)
 31   X(P)=X(P)+DELX(P)
      DO 61 K4=1,NEQ
      T(K4)=TSAV(K4)
 61   CONTINUE
      RETURN
      END


      SUBROUTINE READ(ICELL,IXLIM,IYLIM,NEQ,IMENT,NPERM,IVX,FSENS,ACC)
      COMPLEX S,T
      DIMENSION FREQS(3)
      COMMON/ARRAY1/S(3),T(3),X(50)
      COMMON/ARRAY3/RSAV(50)
      READ(5,1)ICELL,IXLIM,IYLIM,NEQ,IMENT,NPERM,FSENS,ACC
 1    FORMAT(6I10,/,2F10.0)
      IVX=2*NEQ
      READ(5,2) (X(IA),IA=1,IMENT)
 2    FORMAT(1F20.0)
      READ(5,2) (FREQS(IB),IB=1,NEQ)
      DO 3 IC=1,NEQ
      S(IC)=CMPLX(0.0,FREQS(IC))
 3    CONTINUE
      DO 8 IK=1,IMENT
      RSAV(IK)=X(IK)
 8    CONTINUE
      RETURN
      END
```

```
C     THIS SUBROUTINE IS USER SUPPLIED. ONLY THE TRANSFER
C     FUNCTION EXPRESSION TNN) IS CHANGED.

      SUBROUTINE CALCT(N)
      COMPLEX S,T
      COMMON/ARRAY1/S(3),T(3),X(50)
      T(N)=X(5)/(X(1)*X(2)*X(3)*X(4)*X(5)*S(N)**4+X(1)*X(2)*X(3)*S(N)**3
     C+(X(3)*X(4)*X(5)+X(1)*X(4)*X(5)+X(1)*X(2)*X(5))*S(N)**2+(X(3)+X(1)
     C)*S(N)+X(5))
      RETURN
      END


      SUBROUTINE PERM(IVX,JB,NPERM,IPO)
      DIMENSION NV(50,6)
      COMMON/ARRAY5/NVAR(6)
      IPO=IPO+1
      IF(IPO.EQ.1)GO TO 5
      GO TO 6
 5    DO 21 K=1,NPERM
      READ(5,4) (NVAR(JU),JU=1,IVX)
      DO 21 M=1,IVX
      NV(K,M)=NVAR(M)
  21  CONTINUE
 6    DO 7 L=1,IVX
 7    NVAR(L)=NV(JB,L)
    4 FORMAT(5I2)
      RETURN
      END


      SUBROUTINE CENTER(ICELL,IXLIM,IYLIM,NEQ,IF,IG)
      COMPLEX CENT,TNOM
      DIMENSION ANOM(3),RNOM(3),TMUG(3)
      COMMON/ARRAY2/CENT(15,15,3),TNOM(3)
      ID=IXLIM/ICELL
      IE=IYLIM/ICELL
      IF=2*ID+1
      IG=2*IE+1
```

```
      CELL=FLOAT(ICELL)/100.
      DO 6 N=1,NEQ
      ANOM(N)=AIMAG(TNOM(N))
      RNOM(N)=REAL(TNOM(N))
      TMUG(N)=CABS(TNOM(N))
      DO 6 IH=1,IF
      A1=FLOAT(IH)
      A2=FLOAT(IXLIM)/100.0 +CELL
      Y1=A1*CELL*TMUG(N)+RNOM(N)-A2*TMUG(N)
      DO 6 II=1,IG
      A4=FLOAT(II)
      A5=FLOAT(IYLIM)/100.0+CELL
      Z1=A4*CELL*TMUG(N)+ANOM(N)-A5*TMUG(N)
      CENT(IH,II,N)=CMPLX(Y1,Z1)
    6 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE SENSIT(FSENS,IMENT)
      COMPLEX SSAV,SENX,S,DELTAT,T1,T,CDX
      COMMON/ARRAY1/S(3),T(3),X(50)
      COMMON/ARRAY3/RSAV(50)
      COMMON/ARRAY4/SENX(50)
      SSAV=S(1)
      S(1)=CMPLX(0.0,FSENS)
      CALL CALCT(1)
      T1=T(1)
      DO 16 IP=1,IMENT
      DX=1.E-4*RSAV(IP)
      X(IP)=X(IP)+DX
      CDX=CMPLX(0.0,DX)
      CALL CALCT(1)
      DELTAT=T(1)-T1
      SENX(IP)=DELTAT/CDX
      X(IP)=X(IP)-DX
   16 CONTINUE
      S(1)=SSAV
      T(1)=T1
      RETURN
      END
```

```
      SUBROUTINE TWOVAR(ACC,SOLN,IVX,NEQ,ERROR)
      COMPLEX T,SAVT,Z,S
      INTEGER P
      DIMENSION B(50),ERRMUG(500)
      COMMON/ARRAY1/S(3),T(3),X(50)
      COMMON/ARRAY3/RSAV(50)
      COMMON/ARRAY5/NVAR(6)
      COMMON/ARRAY6/Z(3)
      COMMON/ARRAY8/XSAV(50),SAVT(3)
      DO 1 K=1,IVX
      L=NVAR(K)
      B(L)=0.1
C     MAY WANT TO START AT NOMINAL INSTEAD
 1    X(L)=XSAV(L)
      ERRMUG(1)=0.0
      DO 2 M=1,NEQ
      CALL CALCT(M)
 2    ERRMUG(1)=ERRMUG(1)+CABS(T(M)-Z(M))
      ERRSAV=ERRMUG(1)
      MUM=1
C     FOLLOWING IS INITIAL CHOICE OF VARIABLE TO BE INCREMENTED
      MM=1
      P=NVAR(MM)
 216  F=0.1
 201  LUM=1
 215  ICOUNT=1
 203  DO 3 N=1,IVX
      KA=NVAR(N)
 3    X(KA)=X(KA)-F*RSAV(KA)
      MUM=MUM+1
      ICO=ICOUNT
      ICOUNT=ICOUNT+1
      ERRMUG(ICOUNT)=0.0
      DO 4 KB=1,NEQ
      CALL CALCT(KB)
 4    ERRMUG(ICOUNT)=ERRMUG(ICOUNT)+ CABS(T(KB)-Z(KB))
      IF(ERRMUG(ICOUNT).GE.ERRMUG(ICO)) GO TO 202
      ERRSAV=ERRMUG(ICOUNT)
      DO 5 KC=1,IVX
      KD=NVAR(KC)
 5    XSAV(KD)=X(KD)
      DO 6 KE=1,NEQ
 6    SAVT(KE)=T(KE)
      IF(ERRMUG(ICOUNT).LE.ACC)GO TO 10
      GO TO 203
```

```
202   DO 7 KF=1,IVX
      KG=NVAR(KF)
7     X(KG)=X(KG)+F*RSAV(KG)
      IF(ICOUNT.NE.2)GO TO 12
      ICOUNT=1
204   ICO=ICOUNT
      ICOUNT=ICOUNT+1
      DO 8 KH=1,IVX
      KI=NVAR(KH)
8     X(KI)=X(KI)+F*RSAV(KI)
      MUM=MUM+1
      ERRMUG(ICOUNT)=0.0
      DO 9 KJ=1,NEQ
      CALL CALCT(KJ)
9     ERRMUG(ICOUNT)=ERRMUG(ICOUNT)+CABS(T(KJ)-Z(KJ))
      IF(ERRMUG(ICOUNT).GE.ERRMUG(ICO)) GO TO 205
      ERRSAV=ERRMUG(ICOUNT)
      DO 20 KK=1,IVX
      KL=NVAR(KK)
20    XSAV(KL)=X(KL)
      DO 71 KM=1,NEQ
71    SAVT(KM)=T(KM)
      IF(ERRMUG(ICOUNT).LE.ACC) GO TO 10
      GO TO 204
205   DO 82 KN=1,IVX
      KO=NVAR(KN)
82    X(KO)=X(KO)-F*RSAV(KO)
      IF(ICOUNT.NE.2) GO TO 12
      ICOUNT=1
206   ICO=ICOUNT
      ICOUNT=ICOUNT+1
      DO 81 KP=1,IVX,2
      KQ=NVAR(KP)
81    X(KQ)=X(KQ)+F*RSAV(KQ)
      DO  83 KR=2,IVX,2
      KS=NVAR(KR)
83    X(KS)=X(KS)-F*RSAV(KS)
      MUM=MUM+1
      ERRMUG(ICOUNT)=0.0
      DO 84 KT=1,NEQ
      CALL CALCT(KT)
84    ERRMUG(ICOUNT)=ERRMUG(ICOUNT)+CABS(T(KT)-Z(KT))
      IF(ERRMUG(ICOUNT).GE.ERRMUG(ICO)) GO TO 207
      ERRSAV=ERRMUG(ICOUNT)
      DO 85 KU=1,IVX
      KW=NVAR(KU)
85    XSAV(KW)=X(KW)
      DO 86 KX=1,NEQ
```

```
86    SAVT(KX)=T(KX)
      IF(ERRMUG(ICOUNT).LE.ACC)GO TO 10
      GO TO 206
207   DO 87 KY=1,IVX,2
      KZ=NVAR(KY)
87    X(KZ)=X(KZ)-F*RSAV(KZ)
      DO 88 LA=2,IVX,2
      LB=NVAR(LA)
88    X(LB)=X(LB)+F*RSAV(LB)
      IF(ICOUNT.NE.2) GO TO 12
      ICOUNT=1
208   ICO=ICOUNT
      ICOUNT=ICOUNT+1
      MUM=MUM+1
      DO 60 LC=1,IVX,2
      LD=NVAR(LC)
60    X(LD)=X(LD)-F*RSAV(LD)
      DO 61 LE=2,IVX,2
      LF=NVAR(LE)
61    X(LF)=X(LF)+F*RSAV(LF)
      ERRMUG(ICOUNT)=0.0
      DO 62 LF=1,NEQ
      CALL CALCT(LF)
62    ERRMUG(ICOUNT)=ERRMUG(ICOUNT)+CABS(T(LF)-Z(LF))
      IF(ERRMUG(ICOUNT).GE.ERRMUG(ICO)) GO TO 209
      ERRSAV=ERRMUG(ICOUNT)
      DO 63 LG=1,IVX
      LH=NVAR(LG)
63    XSAV(LH)=X(LH)
      DO 64 LI=1,NEQ
64    SAVT(LI)=T(LI)
      IF(ERRMUG(ICOUNT).LE.ACC) GO TO 10
      GO TO 208
 209  DO 66 LJ=1,IVX,2
      LK=NVAR(LJ)
66    X(LK)=X(LK)+F*RSAV(LK)
      DO 67 LL=2,IVX,2
      LM=NVAR(LL)
67    X(LM)=X(LM)-F*RSAV(LM)
      IF(ICOUNT.NE.2) GO TO 12
      IF(F.LE.1.0E-5) GO TO 12
      F=F/10.0
      GO TO 215
15    ICOUNT=1
11    X(P)=X(P)+B(P)*RSAV(P)
      ICO=ICOUNT
      ICOUNT=ICOUNT+1
      IF(LUM.GE.50)GO TO 214
```

```
      LUM=LUM+1
      MUM=MUM+1
      IF(ICOUNT.EQ.20) GO TO 124
      IF(ICOUNT.EQ.50) GO TO 214
119   IF(MUM.GE.200)GO TO 19
      ERRMUG(ICOUNT)=0.0
      DO 401 LN=1,NEQ
      CALL CALCT(LN)
401   ERRMUG(ICOUNT)=ERRMUG(ICOUNT)+CABS(T(LN)-Z(LN))
      IF(ERRMUG(ICOUNT).GE.ERRMUG(ICO)) GO TO 99
      ERRSAV=ERRMUG(ICOUNT)
      XSAV(P)=X(P)
      DO 403 LQ=1,NEQ
403   SAVT(LQ)=T(LQ)
      IF(ERRMUG(ICOUNT).LE.ACC)GO TO 10
      GO TO 11
99    X(P)=X(P)-B(P)*RSAV(P)
      IF(ICOUNT.NE.2)GO TO 12
89    ICOUNT=1
16    X(P)=X(P)-B(P)*RSAV(P)
      ICO=ICOUNT
      ICOUNT=ICOUNT+1
      LUM=LUM+1
      IF(LUM.GE.50) GO TO 214
      MUM=MUM+1
      IF(ICOUNT.EQ.20)GO TO 125
      IF(ICOUNT.GE.50) GO TO 214
117   IF(MUM.GE.200) GO TO 19
      ERRMUG(ICOUNT)=0.0
      DO 404 LR=1,NEQ
      CALL CALCT(LR)
404   ERRMUG(ICOUNT)=ERRMUG(ICOUNT)+CABS(T(LR)-Z(LR))
      IF(ERRMUG(ICOUNT).GE.ERRMUG(ICO)) GO TO 13
      ERRSAV=ERRMUG(ICOUNT)
      XSAV(P)=X(P)
      DO 406 LU=1,NEQ
406   SAVT(LU)=T(LU)
      IF(ERRMUG(ICOUNT).LE.ACC) GO TO 10
      GO TO 16
13    X(P)=X(P)+B(P)*RSAV(P)
      GO TO 12
214   ERRMUG(1)=ERRMUG(ICO)
      GO TO 201
124   B(P)=B(P)*15.0
      GO TO 119
125   B(P)=B(P)*15.0
      GO TO 117
19    SOLN=1.0
```

```
      ERROR=ERRSAV
      RETURN
 12   ERRMUG(1)=ERRMUG(ICO)
      IF(ICOUNT.EQ.2)GO TO 65
      MM=MM+1
      IF(MM.EQ.IVX+1)MM=1
      P=NVAR(MM)
      GO TO 15
 65   B(P)=B(P)/5.0
      GO TO 15
 10   SOLN=2.0
      ERROR=ERRMUG(ICOUNT)
      RETURN
      END




C     SAMPLE DATA DECK FOLLOWS.  DATA IS FOR SAMPLE PROBLEM TWO.
          5        20        20         1         5        10
1.         0.001
1.5307
1.5772
1.0824
0.3824
1.
1.2
 1 2
 1 3
 1 4
 1 5
 2 3
 2 4
 2 5
 3 4
 3 5
 4 5
```

LIST OF REFERENCES

1. Seshu, S. and Waxman, R., "Fault Isolation in Conventional Linear Systems - A feasibility Study," I.E.E.E. Transactions on Reliability, Vol. R-15, p. 11-16. May 1966.

2. Maenpaa, L. H., Stehman, C. J., and Stahl, W. L, "Fault Isolation in Conventional Linear Systems: A Progress Report," I.E.E.E. Transactions on Reliability, Vol. R-18, p. 12-14, February 1969.

3. DeClaris, N., "Driving-Point Impedance Functions of Active Networks," I.R.E. Convention Record, Part 2, p. 26-37, 1956.

4. Phung, L. and Chan, S. G., "On Fault Isolation," Fourth Asilomar Conference on Circuits and Systems, p. 41-45, 1970.

5. Stahl, W. L., Maenpaa, L. H., and Stehman, C. L., "Computer-Aided Design: Part 13 Defining Faults with a Dictionary," Electronics, Vol. 41, p. 64-68, 22 January 1968.

INITIAL DISTRIBUTION LIST

		No. Copies
1.	Defense Documentation Center Cameron Station Alexandria, Virginia 22314	2
2.	Library, Code 0212 Naval Postgraduate School Monterey, California 93940	2
3.	Associate Professor S. G. Chan, Code 52Cd Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	2
4.	Professor R. E. Bach Department of Electrical Engineering Naval Postgraduate School Monterey, California 93940	1
5.	LT Gary L. Long RVAH-3 NAS, Albany, Georgia 31703	1

Security Classification

DOCUMENT CONTROL DATA - R & D

(Security classification of title, body of abstract and indexing annotation must be entered when the overall report is classified)

ORIGINATING ACTIVITY (Corporate author) Naval Postgraduate School Monterey, California 93940	2a. REPORT SECURITY CLASSIFICATION Unclassified 2b. GROUP

REPORT TITLE

Multifault Isolation in Linear Networks by the Method of Joint Signature

DESCRIPTIVE NOTES (Type of report and, inclusive dates)
Master's Thesis; December 1971

AUTHOR(S) (First name, middle initial, last name)

Gary Lamour Long

REPORT DATE December 1971	7a. TOTAL NO. OF PAGES 54	7b. NO. OF REFS 5
8a. CONTRACT OR GRANT NO. b. PROJECT NO c. d.	9a. ORIGINATOR'S REPORT NUMBER(S) 9b. OTHER REPORT NO(S) (Any other numbers that may be assigned this report)	

10 DISTRIBUTION STATEMENT

Approved for public release; distribution unlimited.

11. SUPPLEMENTARY NOTES	12. SPONSORING MILITARY ACTIVITY Naval Postgraduate School Monterey, California 93940

13. ABSTRACT

A study was made to isolate one and two component faults in simple linear networks by the method of joint signature of order two whereby two measurements of a failed network are coded to all possible combinations of two component values which will produce that fault with all other elements at nominal values. Two additional measurements were obtained, a new joint signature of order two generated, and the faulty component set was selected as the fault pair with the same values at both joint signatures. The method involves numerous solutions of sets of nonlinear equations. By assuming many values of a failed network function the equations were presolved on a one time only basis for a network. One and two component faults were simulated in two simple low pass filter networks. The highest success rate in isolating faults was approximately 80%.

KEY WORDS	LINK A		LINK B		LINK C	
	ROLE	WT	ROLE	WT	ROLE	WT
Fault Isolation Signatures						

on
y

Thesis 133899
L793 Long
c.1 Multifault isolation in linear networks by the method of joint signature.